OpenFlare
An Open-Source Web-Security Solution
Phase 2: Work Breakdown

Rayyan Masood,
Ahmar Ayaz,
Ammar Khatri,
Hasan Ali Vejlani,
Abdul Rehman Ansari,
Maleeha
Institute of Business Administration

Abstract

This project aims to enhance network security by designing and implementing a proxy server integrated with a Machine Learning (ML) model and a Redis cache. The proxy server will act as an intermediary for client requests, handling multiple concurrent connections. The ML model, trained on a dataset of network packets, will be used to classify incoming packets as either 'malicious' or 'benign' in real-time. The Redis cache will block IP addresses associated with malicious packets for a certain period. The project excludes client-side implementation and server-side resource management. The system will log all packet data for analysis to identify patterns of malicious activity and improve the ML model. Regular vulnerability assessments will be conducted to ensure system security. The project will be executed by a team of five, with each member responsible for a specific aspect of the project.

1 Inclusions

Design and Implementation of Proxy Server: This involves setting up a server that will act as an intermediary for requests from clients seeking resources from other servers. The server will need to be able to handle multiple concurrent connections and have robust error handling to deal with potential issues that could arise during operation.

Integration of Machine Learning Model: The proxy server will use an ML model to filter out malicious packets. This involves training a model on a dataset of network packets, where each packet is labeled as either 'malicious' or 'benign'. The trained model will then be integrated into the proxy server, where it will be used to classify incoming packets in real-time.

Redis Cache Implementation: A Redis cache will be used to block the malicious IP addresses. When the ML model classifies a packet as malicious, the IP address from which the packet originated will be added to the Redis cache. Any future packets from this IP address will be blocked for a certain period of time. After this time has elapsed, the IP address will be removed from the cache (i.e., flushed out).

2 Exclusions

Client-side Implementation: This project focuses on the server-side implementation, specifically the proxy server, ML model, and Redis cache. It does not include the implementation of client-side applications that will interact with the proxy server.

Server-side Resource Management: The management and maintenance of the resources that the server provides (e.g., databases, web services) are not included in this project. The focus is on the network security aspects.

3 Functional Requirements

Packet Filtering: The system should be able to filter out malicious packets using the ML model. This requires the model to be able to accurately classify packets as either 'malicious' or 'benign'.

IP Blocking: The system should block malicious IP addresses using Redis cache. This requires the cache to be able to quickly add and remove IP addresses.

Packet Forwarding: The system should forward valid packets to the server. This requires the proxy server to have a high throughput, as it needs to handle all incoming and outgoing traffic.

4 Hardware and Software Requirements

Hardware: A server machine with sufficient processing power and memory to handle the ML model and Redis cache. The exact specifications will depend on the

expected network traffic and the complexity of the ML model.

Software: A suitable operating system for running the server (e.g., Linux). Docker for containerization. Redis for caching.
A suitable programming language and framework for implementing the proxy server and integrating the ML model (e.g., Python with Flask)

5 Vulnerability and Data Analysis Methodology

Vulnerability Assessment: Regular vulnerability assessments should be conducted to identify any potential weaknesses in the system. This could involve techniques such as penetration testing or static code analysis.

Data Analysis: The system should log all incoming and outgoing packets for analysis. The data can be analyzed to identify patterns of malicious activity and improve the ML model. This could involve techniques such as data mining or machine learning.